

Keep in Touch with EmailJS

By Alex Wilson

Hello, I'm Alex!

You can visit (and send me emails from) my website: CodeByAlex.com

Follow me on Twitter at: [@CodeByAlex](https://twitter.com/CodeByAlex)

Dive into the code on GitHub at: [@CodeByAlex](https://github.com/CodeByAlex)

I will also be at the bar after the presentation if you have any questions or comments

Use an Email Service or Create Your Own

When to use a service:

- You want to **abstract** away email logic
- You don't want to manage the **security** aspect of sending emails
- You no longer want to pay hosting **fees** to run your email service

When to create your own service:

- You're interested in **learning** how to send emails from scratch
- You need **features** that a pre made service does not provide
- You want full **control** over how the emails are being sent

My Motivation and Requirements

Move away from my developed solution created for my portfolio website, consisting of Node and a Mongo DB hosted on an EC2 instance, that I would have to continue to maintain and pay for hosting

Requirements to meet my goal:

- The service should be relatively cheap or free
- The service should provide a success or failure response upon submitting the email
 - This would allow follow up animations to be triggered
- Records should be kept about when emails are sent and who they are sent from
- No difference should be seen by the user after the switch to a new service

Service Option: FormSpree

Via Code Snippet

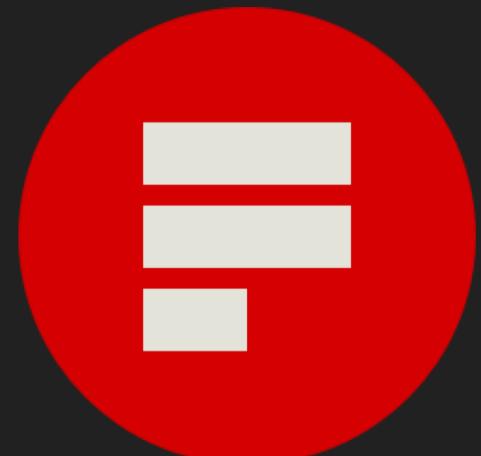
- Releases personal email via the code snippet that can be added to the codebase

EXAMPLE FORM

```
<form action="https://formspree.io/your@email.com" method="POST">
  <input type="text" name="name">
  <input type="email" name="_replyto">
  <input type="submit" value="Send">
</form>
```

Via API call

- Requires a captcha if trying to use as an API and does not keep record of your emails unless you upgrade to their “Gold” plan



Service Option: SendGrid

Pros:

- Allows for 100 emails to be sent per day as a part of the free plan
- Maintains a record of all emails sent to you
- Provides visuals displaying your SendGrid usage statistics
- Supplies plenty of documentation for several different languages

Cons:

- Prefers if you make calls from your own server and will throw a security error if you try to make a call from a client side application



Service Option: EmailJS

- Pros
 - 3 different email templates can be set up with their free tier
 - A record will be kept of all email interactions
 - Pure JavaScript can be used to send emails via their SDK
 - A rest endpoint is provided as an alternative to their SDK
 - No personal email information will be exposed to the client

- Cons
 - Allows for only 200 monthly emails with their free tier



How to use EmailJS in 5 Steps

1. Add an email Service to your EmailJS Account
2. Create a template through the EmailJS Website
3. Develop an HTML form
4. Access the form values
5. Send the email to an EmailJS endpoint containing form information and EmailJS Credentials

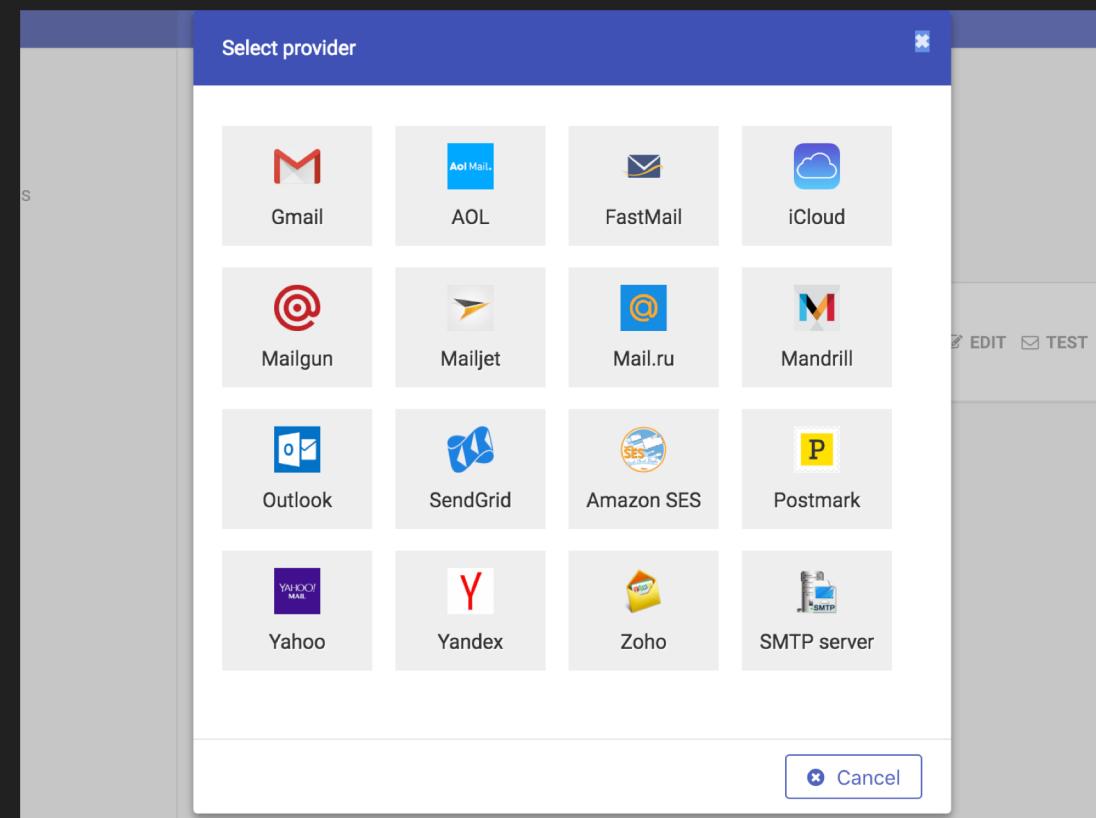
Create an EmailJS Email Service

- Go to the email services tab

The screenshot shows the 'Email Services' section of the EmailJS dashboard. On the left, there's a sidebar with icons and labels: 'Email Services' (selected), 'Email Templates', 'Contacts', 'Email History', 'Logs', 'Statistics', 'Installation', and 'Account'. The main area has a title 'Email Services' and a blue button '+ Add new service'. Below that, there's a card for a 'Gmail' service. The card features the Gmail logo, the text 'Gmail' and 'Service ID: gmail', a 'Default' button, and three small icons for 'EDIT', 'TEST', and 'DELETE'.

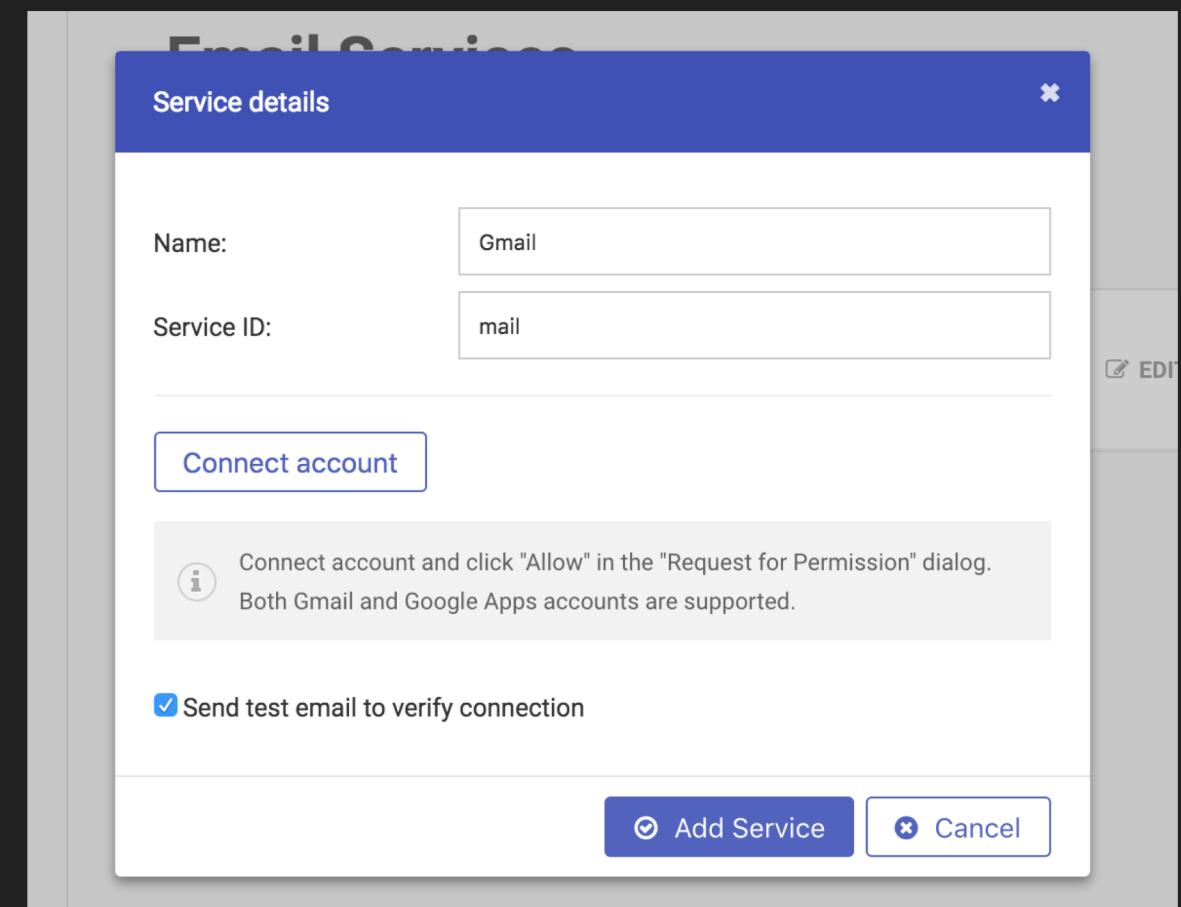
Create an EmailJS Email Service

- Select your email service



Create an EmailJS Email Service

- Create a name and Service ID that will be associated with this email service
- Click the "Connect account" button and login to the email where you would like the emails to be sent
- Click the "Add Service" button to complete the process



Setting Up Your Email Template

- Go to the Email Templates tab

The screenshot shows a user interface for managing email templates. On the left, there is a sidebar with the following navigation options:

- Email Services
- Email Templates** (selected)
- Contacts
- Email History
- Logs
- Statistics
- Installation
- Account

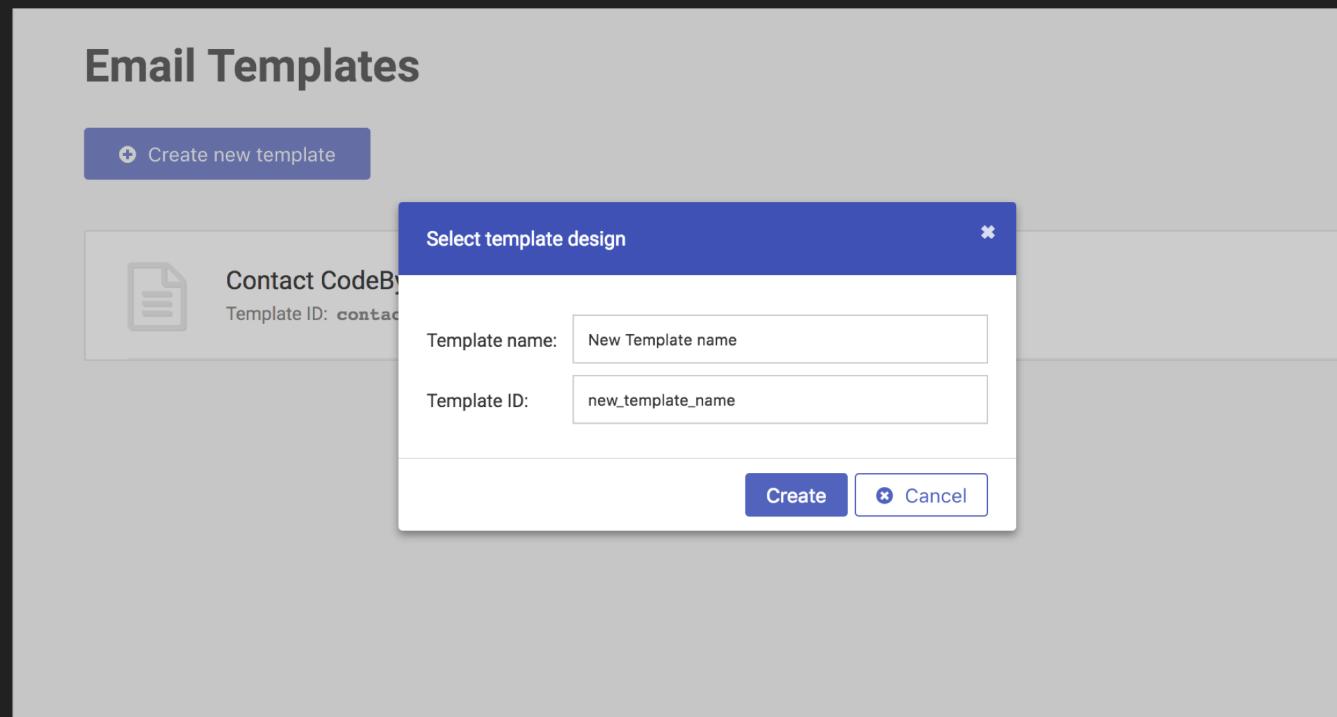
The main content area is titled "Email Templates". It features a prominent blue button labeled "Create new template". Below this, a specific template is listed:

Contact CodeByAlex
Template ID: EDIT CLONE DELETE

The "Email Templates" section has a light gray background, while the sidebar and other content areas have a white background.

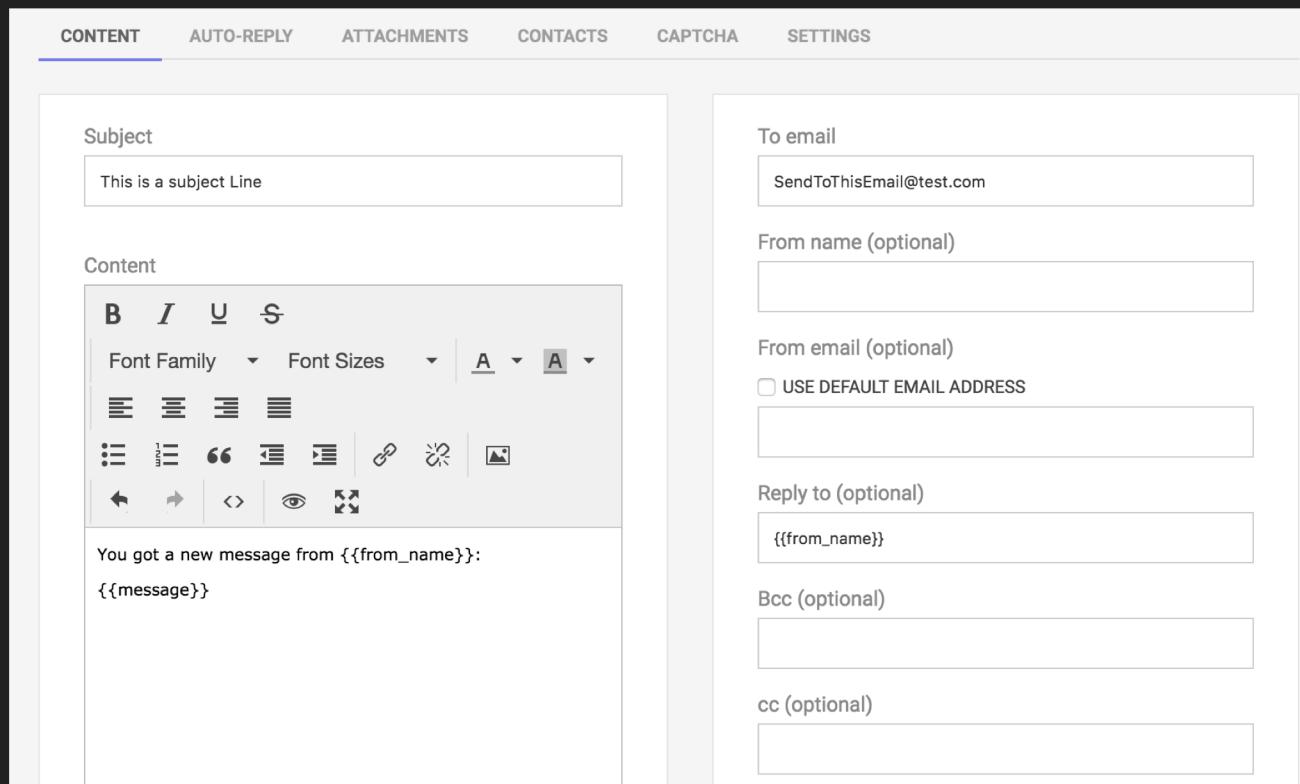
Setting Up Your Email Template

- Create new Template



Setting Up Your Email Template

- Add Template variables and choose email options



Creating Your HTML Form

- Develop a form using html with inputs corresponding to the variable names you have set up in your template
- Include the minified EmailJS Javascript file
- Initialize your EmailJS User using the user id provided upon registration

```
1 ▼ <form id="myform">
2
3 ▼   <label>from_name</label>
4   <input type="text" name="from_name" />
5
6 ▼   <label>message</label>
7   <input type="text" name="message" />
8
9   <br><br>
10 ▼  <button> Send </button>
11
12 </form>
13
14
15 <script type="text/javascript"
16   src="https://cdn.emailjs.com/sdk/2.2.4/email.min.js"></script>
17
18 ▼ <script type="text/javascript">
19   (function(){
20     emailjs.init(USER_ID);
21   })();
22 </script>
```

Connecting Your Application to EmailJS

- Get access to your form object
- On submit of that form
 - Retrieve form values
 - Use the emailjs.send function, by providing the service and template IDs as well as the form values
 - In the promise, provide response actions that will be triggered upon success or failure
- As a user, click the submit button and check your email!

```
1  const myform = $("form#myform");
2  myform.submit(function(event){
3      event.preventDefault();
4
5      // get values from the form and place them into a params variable
6      const params = myform.serializeArray().reduce(function(obj, item) {
7          obj[item.name] = item.value;
8          return obj;
9      }, {});
10
11     const service_id = "default_service";
12     const template_id = "new_template_name";
13
14     emailjs.send(service_id,template_id,params)
15     .then(function(){
16         alert("Sent!");
17     }, function(err) {
18         alert("Send email failed!\r\n Response:\n " + JSON.stringify(err));
19     });
20     return false;
21 });
```

Demo

More Information

- EmailJS: <https://www.emailjs.com>
- FormSpree: <https://formspree.io>
- SendGrid: <https://sendgrid.com>
- My Angular/TypeScript implementation: <https://github.com/CodeByAlex/Code-By-Alex-Front-End/blob/master/src/app/contact/email.service.ts>